

**PATENT****IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of:

Jeremy Stein Cohen, et al.

Application No. 09/912,280

Filed: 23 July 2001

For: *System and Method for Analyzing  
Transaction Data*

Group Art Unit: 2173

Examiner: Raymond J. Bayeri

Attorney Docket No.: 00982.0004.NPUS00

**DECLARATION PURSUANT TO 37 C.F.R. §1.131****BOX ISSUE FEE**Assistant Commissioner for Patents  
Washington D.C. 20231

Sir:

I, Ashok Srivastava, do hereby declare as follows:

1. I am a co-inventor of the above-identified application. I received Ph.D., MS, and BS degrees in Electrical Engineering from the University of Colorado at Boulder. I was employed at Blue Martini from Feb 14, 2000 to August 18, 2002. At Blue Martini, I was a Senior Director. I am currently employed at NASA Ames Research Center.

2. I contributed to the conception and reduction to practice of the above-identified application. I was involved with the entire process of filing the patent application.

3. I reviewed the newly discovered reference U.S. Patent 6,684,206, which was issued on January 27, 2004. The patent was filed May 18, 2001. The patent does not claim priority to any other patent or application.

**BEST AVAILABLE COPY**

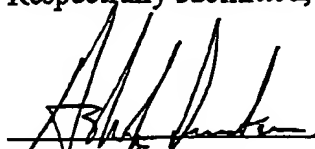
Serial No. 09/912,280  
Attorney Docket No. 00982.0004.NPUS00

4. The present invention was conceived prior to May 18, 2001 and was coupled with due diligence from prior to May 18, 2001 to the filing of the application on July 23, 2001. A copy of a computer program created while at Blue Martini is enclosed herewith; the non-relevant but proprietary portion of the record is redacted. The attached program is from before May 18, 2001. The attached program shows that the inventors were performing work on a node focus matrix, a part of COLAP prior to May 18, 2001. From May 18, 2001 until July 23, 2001, I and/or other co-inventors including Jeremy Cohen exercised due diligence and worked with the patent attorneys at Howrey Simon Arnold & White to file the application. During this period, I and/or other co-inventors including Jeremy Cohen also were continuing to test and implement the invention.

5. I declare further that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that the making of willful false statements and the like are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful statements may jeopardize the validity of the applications or any patent issuing thereon.

Respectfully submitted,

Dated: 2004/1/02

  
\_\_\_\_\_  
Ashok Srivastava

BEST AVAILABLE COPY

```

%-----
%-----
%-----
%-----
%-----
%Function Name: COLAP
%
%Function Call: COLAP
(clientName,Click_Lines,TheNodeNames,NumForwardSteps,NumBackwardSteps,InitialTargetNodeShowLifts)
%
%Author: Jeremy Cohen
%Original Date:
%
%Description: Launches the Clickstream OLAP tool for investigating
%              clickstream patterns visually.
%
%Modified By:
%Modified Date:
%Modifications:
%
%-----
%-----
%-----
%INPUT REQUIREMENTS: [optional]
%
%              clientName: a string with the name of the client (may be
%              '');
%              Click_Lines (Nx(M>=3)): matrix of clicks in sequential
order (based on Mine_Click_Lines in CIS/DSS)
%              - must contain:
%              - SSN_ID in column 1 = session identifier
%              - Seq_Num in column 2 = sequence # of
session
%              - Node_ID in column 3 = lowest level node
id
%              - these must be numbered starting
from 1 by 1's
%
%              - may contain:
%              - Node_category
%              - assortment_id associated with this click
%              - product_id associated with this click
%              - others
%
%              TheNodeNames: a cell vector such that TheNodeNames(i) =
the name of the node with node_id=i in Click_Lines
%
%              [NumBackwardSteps]: the number of clicks to look backward
initially (default = 3)
%              [NumForwardSteps]: the number of clicks to look forward
initially (default = 3)
%              [InitialTargetNode]: the initial target node id (default=1)
%              [ShowLifts]: 1 to show lifts vs. entire population, 0 to
show straight probabilities (default = 0);
%
%
%
%OUTPUT:

```

```

%
%
%MODIFIES:
%          COLAP will close any open figures.
%
%OTHER INSTRUCTIONS:
%
%-----
%-----
%-----
%Notes/Issues:
%          %have to set target node in command line if not shown on plot
%Known Bugs:
%          % will fail if NumBackwardSteps > -1 and focus = ENTER
%          % will fail if NumForwardSteps < 1 and focus = EXIT
%
%Potential Enhancements:
%          (1) allow collapse of time (0-10) (11-20) (21-30)
%          (2) allow filters on session criteria (did they buy,
customer is male, longer than 10 clicks, ...)
%          (1) filter out short paths (and produce report on them)
%          (2) see if weblogs can be used
%          (3) Code Review
%          (4) enable filter by discrete session characteristic
%          (5) think about more automatic "pattern searches" like:
%              'low dispersion -> high dispersion -> low dispersion'
%          (6) hierarchical time dimension (collapse steps 0-10, 11-
20, 21-30...)
%          (7) colors on plot: lift?, proximity to exit?,
demographics?
%
%See also:
%
%-----
%-----
%----- BLUE MARTINI SOFTWARE -----
E-BIS --
%-----
%-----

```

```

function COLAP
(clientName, Click_Lines, TheNodeNames, NumForwardSteps, NumBackwardSteps, In
itialTargetNode, ShowLifts)

```

```

a.clientName = clientName;
a.TheNodeNames = TheNodeNames;
a.Click_Lines = Click_Lines;

```

```

%filter out clickstreams of length <=5
% DEBUG: added on 7/25
%*****%
%short_session_thresh = 5;
status = 'Building Session Cube'
a.SessionStats = unique_count(Click_Lines(:,1));
%short_session_list = a.SessionStats(find(a.SessionStats(:,2) >
short_session_thresh),1);
%a.Click_Lines_Short_Sessions = Click_Lines(find(~ismember(Click_Lines
(:,1), short_session_list)),:);
%a.Click_Lines = Click_Lines(find(ismember(Click_Lines
(:,1), short_session_list)),:);

```

```

%a.SessionStats = a.SessionStats(find(a.SessionStats(:,2) >
short_session_thresh),:);
a.SessionPathLength = zeros(max(max(a.SessionStats(:,1))),1);
for j=1:length(a.SessionStats)
    if a.SessionStats(j,2) < 10
        a.SessionPathLength(a.SessionStats(j,1)) = 2;
    else
        if a.SessionStats(j,2) < 31
            a.SessionPathLength(a.SessionStats(j,1)) = 3;
        else
            a.SessionPathLength(a.SessionStats(j,1)) = 4;
        end
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

switch nargin
case 7
    a.NumForwardSteps = NumForwardSteps;
    a.NumBackwardSteps = NumBackwardSteps;
    a.InitialTargetNode = InitialTargetNode;
    a.ShowLifts = ShowLifts;
case 6
    a.NumForwardSteps = NumForwardSteps;
    a.NumBackwardSteps = NumBackwardSteps;
    a.InitialTargetNode = InitialTargetNode;
    a.ShowLifts = 0;
case 5
    a.NumForwardSteps = NumForwardSteps;
    a.NumBackwardSteps = NumBackwardSteps;
    a.InitialTargetNode = 1;
    a.ShowLifts = 0;
case 4
    a.NumForwardSteps = NumForwardSteps;
    a.NumBackwardSteps = 3;
    a.InitialTargetNode = 1;
    a.ShowLifts = 0;
case 3
    a.NumForwardSteps = 3;
    a.NumBackwardSteps = 3;
    a.InitialTargetNode = 1;
    a.ShowLifts = 0;
end

%%% Error checking

%Call the script RunNodeFocus2 with the appropriate data
a.tgt_matrix = [2 0 3 a.InitialTargetNode 0];

% tgt_matrix format:
%   Column 1 specifies phrase type:
%   {COL1}==1 ----> Click_Lines(r+ {COL2},{COL3}) ==
Click_Lines(r+{COL4},{COL5});
%   {COL1}==2 ----> Click_Lines(r+ {COL2},{COL3}) == {COL4});
%   {COL1}==3 ----> Click_Lines(r+ {COL2},{COL3}) ==
Click_Lines(r+{COL4},{COL5});

global global_A global_G;

```

RunNodeFocus2(a);